HeatFlex: Machine learning based data-driven flexibility prediction for individual heat pumps

Brusokas, Jonas; Pedersen, Torben Bach; Siksnys, Laurynas; Zhang, Dalin; Chen, Kaixuan

Published in:
e-Energy '21: Proceedings of the Twelfth ACM International Conference on Future Energy Systems

DOI (link to publication from Publisher):
10.1145/3447555.3464866

Publication date:
2021

HeatFlex: Machine learning based data-driven flexibility prediction for individual heat pumps

Jonas Brusokas
Aalborg University
jonasb@cs.aau.dk

Torben Bach Pedersen
Aalborg University
tbp@cs.aau.dk

Laurynas Šikšnys
Aalborg University
siksnys@cs.aau.dk

Dalin Zhang
Aalborg University
dalinz@cs.aau.dk

Kaixuan Chen
Aalborg University
kchen@cs.aau.dk

ABSTRACT

With their rising adoption and integration into smart grids, heat pumps are becoming an increasingly important source of flexible energy. Heat pump flexibility can be utilized by using controllers to remotely manage their operation while maintaining the temperature within predefined user comfort bounds. Traditional indoor temperature modelling approaches require detailed information about the deployment site, device specific parameters and monitored data, making them inapplicable for the majority of heat pump deployments. This paper proposes a novel data-driven machine learning based method HeatFlex for indoor temperature forecasting and flexibility prediction using only 3 monitored variables: indoor and outdoor temperatures and heat pump power consumption. HeatFlex enables plug-and-play flexibility prediction from heat pumps without requiring exact device and building specifications or installation of additional sensors. This paper also introduces novel flexibility metrics enabling quantitative evaluation of heat pump flexibility prediction performance. HeatFlex is based on deep learning predictive models: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) recurrent neural networks. Our experimental evaluation compared these networks with traditional multivariate linear regression and SARIMAX time series forecasting model baselines. HeatFlex performance was qualitatively and quantitatively evaluated using data from three real-world heat pump deployments with different building sizes, heat pump types and specifications. Experimental results indicate that HeatFlex is effective to accurately predict over 90% of available potential flexibility.

CCS CONCEPTS

• **Applied computing** → **Forecasting**; • **Computing methodologies** → *Neural networks*; *Supervised learning*.

KEYWORDS

flexibility prediction, indoor temperature forecasting, smart grid ready heat pump, machine learning, deep learning

e-Energy '21, June 28–July 2, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8333-2/21/06.
https://doi.org/10.1145/3447555.3464866

ACM Reference Format:
Jonas Brusokas, Torben Bach Pedersen, Laurynas Šikšnys, Dalin Zhang, and Kaixuan Chen. 2021. HeatFlex: Machine learning based data-driven flexibility prediction for individual heat pumps. In *The Twelfth ACM International Conference on Future Energy Systems (e-Energy '21), June 28–July 2, 2021, Virtual Event, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447555.3464866

1 INTRODUCTION

Over the past few years, significant advancements were made in the development of smart grids, capable of balancing supply and demand and maximising the utilization of renewable energy sources (RES). The majority of recent research focuses on flexibility based demand response mechanisms for shifting energy supply and demand according to the availability of RES and other factors. Previous research has identified and built on various aspects of energy flexibility, such as: aggregation and disaggregation of flexibility [29], frameworks and architectures for flexibility data management [4, 25], flexibility modeling [26], measurement and comparison [33], and trading in energy markets [34].

Heat pumps have seen a significant rise in adoption in Europe [6]. They offer an attractive alternative to conventional electrical heating solutions due to their substantially higher efficiency and, therefore lower carbon footprint [11, 22, 35]. With heating constituting the largest part of energy demand for both residential and commercial sectors, heat pump installations are key sources of potential energy flexibility [23]. One way of harnessing potential flexibility is allowing deviations in building indoor temperature within predefined user comfort bounds. The heat pumps could then be controlled to increase or decrease power consumption for a period of time while still maintaining indoor temperature within given bounds [23]. In order to utilize available flexibility a remote control mechanism is required which would allow dynamic modification of heat pump operation. Also, an indoor temperature forecasting model would be required to estimate how the modified operation will impact indoor temperature to make sure it does not violate the user defined comfort bounds.

In recent years, an interface specification known as Smart Grid Ready (SG-Ready) has been developed enabling heat pump operation to be adjusted using an external signal. It has seen widespread adoption among manufacturers and is available in more than 1200 heat pump models [11]. As seen in Table 1, SG-Ready defines 4 supported operating modes. Out of the 4 specified operation modes, *Off* and *ForcedOn* enforce deterministic, implementation-agnostic

heat pump behaviour. Mode *Off* interrupts heating and reduces heat pump power consumption to a minimum, where as *ForcedOn* forces the heat pump to consume the maximum amount of power, increasing the heat. These two modes enable the utilization of heat pump flexibility, as they allow to temporarily decrease power consumption with mode *Off* or increase power consumption with *ForcedOn* mode.

Table 1: SG-Ready heat pump operating modes

Operating mode	Definition
Off	Heat pump operation actively switched off, using minimal power (limited to 2 hours a day)
Normal	Heat pump operation is normal
RecommendedOn	Heat pump operation is set to prefer switching on (interpreted as a recommendation)
ForcedOn	Heat pump (and auxiliary heaters, if applicable) operation actively switched on, using maximum power

However, there are no known scalable, data-driven methods for individual smart heat pump flexibility prediction that estimate what heat pump operation is possible given user predefined temperature comfort bounds. Previously conducted studies utilize physical models to forecast thermal response, however, these methods require exact heat pump specifications, detailed information about the building and its layout which are unique to each deployment [23]. Furthermore, these methods use a lot of monitored variables which are not always available due to lack of sensors [15, 24].

This paper aims to address these limitations by proposing a novel data-driven flexibility prediction method HeatFlex for individual smart heat pumps and makes the following contributions: (1) Proposes HeatFlex, the first, to the authors' knowledge, fully data-driven flexibility prediction method for smart heat pumps based on modern machine learning methods. (2) Introduces novel flexibility metrics, enabling quantitative evaluation of predicted flexibility. (3) Experimental qualitative and quantitative evaluation of HeatFlex using real world heat pump data from three deployments. HeatFlex is fully data-driven and does not require a predefined physical model of the deployment environment or exact specifications of the heat pump and only uses 3 monitored variables for prediction: indoor temperature, heat pump power consumption and outdoor temperature, which are widely available from most modern energy management systems and open-access weather information providers. Our experimental evaluation shows that HeatFlex, utilizing modern recurrent neural networks, significantly outperforms multivariate linear regression and SARIMAX time series forecasting model baselines in terms of prediction error. Results show that HeatFlex is capable of accurately predicting over 90% of available flexibility throughout different seasons and heat pump deployments.

The paper is structured as follows: Section 2 describes related work. Section 3 defines the flexibility prediction and indoor temperature forecasting problems. Section 4 provides an outline of the proposed method HeatFlex for flexibility prediction. Section 5 describes used indoor temperature forecasting models and metrics. Section 6 describes our proposed algorithm for predicting flexibility and proposed flexibility metrics. Section 7 describes setup for the experiments. Section 8 shows the results of the experiments. Finally, Section 9 concludes the paper and outlines directions for future work.

2 RELATED WORK

In previous work, several methods have been proposed for flexibility prediction. A regression and temporal sequence matching based method has been proposed for predicting device activation time and energy demand of electric vehicles and wet-devices (washing and cleaning devices that use water, like dishwashers or washer dryers) [20, 21] . A physical model based energy demand prediction algorithm has been proposed for shifting demand of electrical vehicles and heat pumps to minimize the cost of consumed electricity [23]. Another study proposes a physical modelling approach for estimating the energy profile of a building with an installed heat pump expressed as a first order linear time invariant system [21]. In comparison, this paper proposes HeatFlex, a model-free, fully data-driven method for predicting flexibility from individual smart heat pumps.

Traditionally, physical modelling through parametrized differential equations was used to model building's thermal response to change in heating operation of HVAC systems, including heat pumps [23]. However, this approach requires prior knowledge about the heating installations, building layout, precise device specifications and a significant number of monitored variables through installed sensors, which are not typically available in real world deployments. Furthermore, the physical modelling approach requires defined equations describing the exact environment configuration for each individual deployment and is also heavily reliant on constantly available high frequency, accurate sensor readings [15, 24]. Alternative data-driven methods, such as regression methods or time-series forecasting models do not have such strict prerequisites and have been applied to solve load forecasting problems [10]. However, regression methods do not capture temporal dependencies and most cannot model non-linear relationships, whereas time series forecasting models tend to underperform on highly non-linear, non-stationary data [18, 30]. HeatFlex utilizes modern data-driven modeling methods that can capture both sequential and complex, non-linear relationships.

Over the past few years, machine learning based approaches have been successfully adopted and achieved state-of-the-art performance for various prediction problems in the energy domain [2, 12, 14]. Modern machine learning methods like deep neural networks are capable of modelling complex, non-linear relationships between variables, having much greater applicability and, usually, higher performance than traditional regression or time series forecasting algorithms [24]. Recently, these methods have been applied for load and thermal response prediction in buildings equipped with HVAC systems. Traditional neural networks have been used for single step ahead humidity and indoor temperature forecasting in households [31], indoor temperature forecasting for several areas of a school building [1]. More advanced recurrent

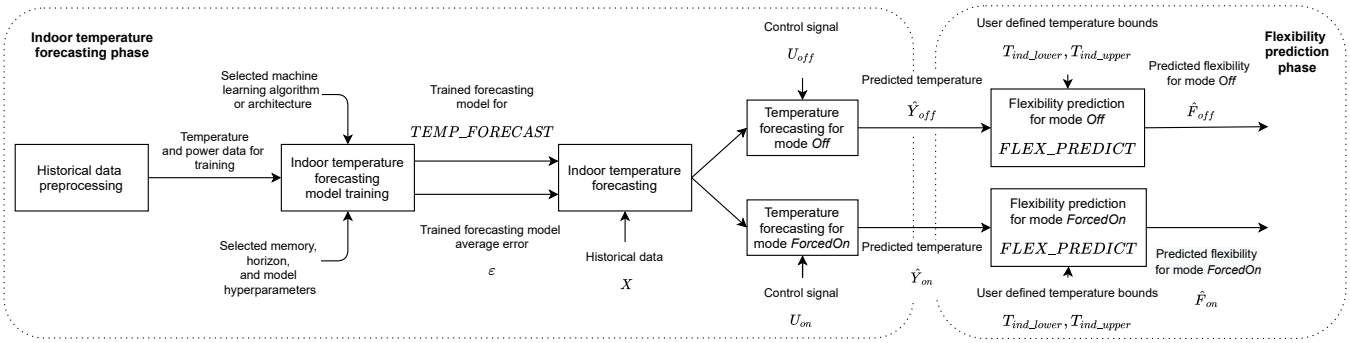

Figure 1: Overview of HeatFlex: data-driven smart heat pump flexibility prediction method

neural networks, such as the Long-Short Term Memory (LSTM) network have also been used for indoor temperature forecasting in a building with an installed air handling unit [17], outperforming traditional neural network models. LSTM based models have also been used for heating load forecasting for combined heating and power plants, yielding significant performance gains compared to traditional regression approaches [18]. This paper improves on previous research by effectively applying modern neural networks for smart heat pump indoor temperature forecasting and flexibility prediction.

3 PROBLEM DEFINITION

This section formally describes the preliminaries and provides a problem definition for the paper.

$$X = \begin{bmatrix} X(t-m+1) & \dots & X(t-1) & X(t) \end{bmatrix} \quad (1)$$

X is a regular time series holding historical data of past observed parameter values with record count m (known as memory) defining how many past records will be used for prediction. X is composed of vectors $X(t)$:

$$X(t) = \begin{bmatrix} T_{ind}(t) & T_{out}(t) & P_{hp}(t) & DT(t) \end{bmatrix} \quad (2)$$

where $T_{ind}(t)$ is the indoor temperature in a selected environment at time t, $T_{out}(t)$ is the outdoor (external) temperature at time t, $P_{hp}(t)$ is the average power consumption by the heat pump in the time interval $(t-s;t]$, and $DT(t)$ is the date and time at which $T_{ind}(t)$, $T_{out}(t)$, and $P_{hp}(t)$ were gathered. Variables $T_{ind}(t)$ and $P_{hp}(t)$ can be retrieved from the energy management system of the heat pump, variable $T_{out}(t)$ can be retrieved from a locally installed sensor or an open access weather service, and $DT(t)$ can be retrieved along with the other variables or be generated.

$$U = \begin{bmatrix} P_{hp}(t+1) & \dots & P_{hp}(t+h-1) & P_{hp}(t+h) \end{bmatrix} \quad (3)$$

U is a one-dimensional vector of length h called the control signal. It contains planned heat pump power consumption values $P_{hp}(t+i)$, where $i <= h$. h is known as prediction horizon and defines how many time steps of indoor temperature will be predicted by the model.

$$Y = \begin{bmatrix} T_{ind}(t+1) & \dots & T_{ind}(t+h-1) & T_{ind}(t+h) \end{bmatrix} \quad (4)$$

Y and $\hat{Y}$ are one-dimensional vectors of length h, containing ground-truth and predicted indoor temperatures, respectively.

In this paper heat pump flexibility is denoted as F. Flexibility $F \in \mathbb{Z}, F > 0$ is a positive integer scalar, representing how many timesteps the user predefined lower temperature bound T_{ind_lower} and upper temperature bound T_{ind_upper} will not be violated if the heat pump is operating according to the control signal U. Predicted flexibility is defined as $\hat{F}$, whereas existing potential flexibility is denoted as F.

Problem definition. Let *FLEX_PREDICT* be a function taking predicted indoor temperature $\hat{Y}$, lower temperature bound T_{ind_lower} and upper temperature bound T_{ind_upper}, returning predicted flexibility $\hat{F}$, such that:

$$FLEX\_PREDICT : (\hat{Y}, T_{ind\_lower}, T_{ind\_upper}) \mapsto \hat{F} \quad (5)$$

Let *TEMP_FORECAST* be a function taking historical readings X, control signal U, returning forecasted temperature $\hat{Y}$, such that:

$$TEMP\_FORECAST : (X, U) \mapsto \hat{Y} \quad (6)$$

Given historical data X, control signal U, lower user defined temperature bound T_{ind_lower} and upper user defined temperature bound T_{ind_upper}, find the two functions *FLEX_PREDICT* and *TEMP_FORECAST*, such that prediction errors for predicted flexibility $|\hat{F} - F|$ and forecasted indoor temperature $|\hat{Y} - Y|$ are minimized.

4 HEATFLEX OVERVIEW

This section outlines the flexibility prediction method HeatFlex proposed in this paper. HeatFlex utilizes data-driven predictive models to learn *TEMP_FORECAST*, an indoor temperature forecasting function used to accurately forecast indoor temperature $\hat{Y}$ given historical data X and control signal U for prediction horizon h. As seen in Figure 1, HeatFlex has two phases: indoor temperature forecasting and flexibility prediction.

For the indoor temperature forecasting phase, the historical X and U data are preprocessed (see Section 7.2). Afterwards, the machine learning algorithm for the temperature forecasting model (see Section 5) and its parameters are selected, including memory parameter m, horizon parameter h and model-specific hyper-parameters (such as cell count for neural networks). Once the model is selected, the preprocessed historical data is used to train the selected model to predict indoor temperature $\hat{Y}$, with the training goal of minimizing error $|\hat{Y} - Y|$. The trained model is then used to forecast indoor temperature $\hat{Y}$ in two scenarios. The first scenario *Off* assumes that the heat pump will be switched to mode *Off*, interrupting heating.

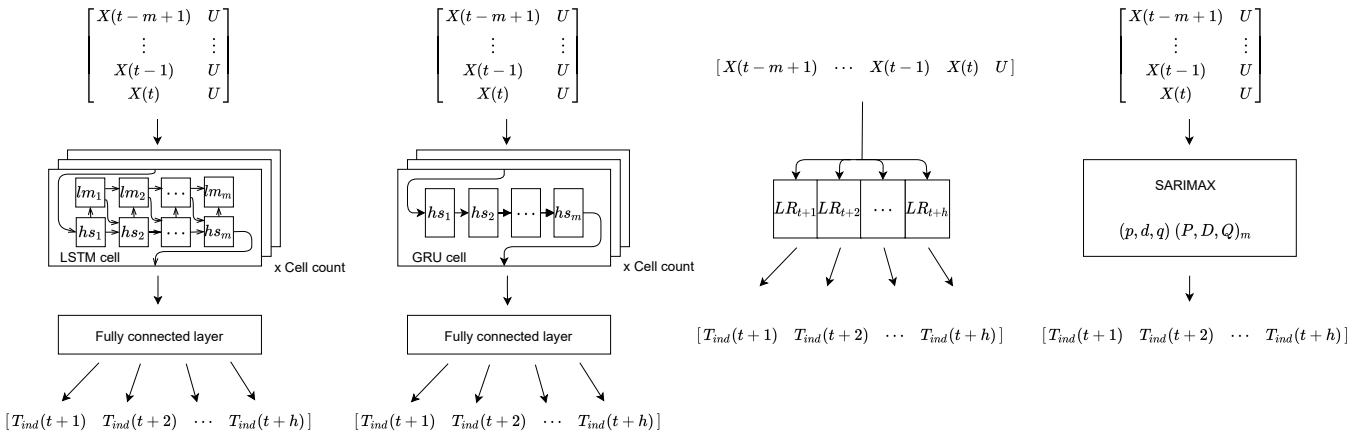

Figure 2: Overview of used indoor temperature forecasting models (visualized in order starting from the left: LSTM, GRU, Multivariate Linear Regression, SARIMAX)

In this case, the trained model will take historical data X and control signal U_{off} as inputs and predict how the temperature will change, returning predicted temperature $\hat{Y}_{off}$. In the second scenario *ForcedOn*, the same procedure is repeated for mode *ForcedOn*, using historical data X and control signal U_{on} to predict $\hat{Y}_{on}$.

During the flexibility prediction phase, the forecasted temperatures are used to predict flexibility. In both scenarios *Off* and *ForcedOn*, the predicted temperature along with user defined temperature bounds T_{ind_lower}, T_{ind_upper} are used to predict flexibility, using function *FLEX_PREDICT* as described in Section 6. In scenario *Off*, $\hat{Y}_{off}$ is used to predict $\hat{F}_{off}$, whereas in scenario *ForcedOn*, $\hat{Y}_{on}$ is used to predict $\hat{F}_{on}$. The average forecasting error ε calculated during model training can be optionally used in *FLEX_PREDICT* to adjust for forecasting error when predicting flexibility (see Section 6).

5 INDOOR TEMPERATURE FORECASTING

In this paper, predictive models are used to learn the function *TEMP_FORECAST* to accurately forecast indoor temperature $\hat{Y}$ for horizon h, given X and U. Metrics are used to quantitatively evaluate prediction accuracy.

5.1 Predictive models

Four different models were analyzed for indoor temperature forecasting. These include two modern recurrent neural networks: Long Short-Term Memory, Gated Recurrent Unit and two baseline models: Multivariate Linear Regression, Seasonal Auto-Regressive Integrated Moving Average with eXogenous variables (SARIMAX). As shown in Figure 2, all models apart from linear regression use concatenated matrices X and U as inputs, where U is repeated for each time step in matrix X. Because linear regression models cannot handle sequential data, inputs for them are formed by flattening matrix X and then concatenating with U.

Long Short-Term Memory recurrent neural networks. These are the most commonly utilized recurrent neural networks for sequential data. Long Short-Term Memory (LSTM) was introduced to solve the original recurrent neural network's (RNN) inability to capture long-term non-linear dependencies in sequential data [3]. The LSTM architecture utilizes an improved hidden unit, called memory cell, which calculates hidden state hs_t and long term memory lm_t from previously seen sequential data. The memory cell has parameters which define how the cell attains, updates and discards information from hs_t and lm_t, which get learned during training. As seen in Figure 2, during inference, hs_i and lm_i are recursively calculated using input vector x_i from concatenated matrices X and U at time step $i \in [1, m]$ and previously calculated hidden state hs_{i-1} and long term memory lm_{i-1} [13]. As seen in Figure 2, an LSTM neural network uses multiple memory cells (denoted by c). The output of the network is calculated by using a fully connected layer, calculating a weighted sum of the final values of hs from all of the cells for each value in output vector $\hat{Y}$.

Gated Recurrent Unit recurrent neural networks. These networks were introduced as a potential improvement to the LSTM neural network [8, 9]. Like the LSTM, GRU uses a memory cell to calculate a hidden state from previously seen sequential data. The key difference is that GRU does not use two separate hidden states hs_t and lm_t, instead using only one: hs_t. This change results in a simplified structure of the memory cell and a slight decrease in required computation. As shown in Figure 2, the GRU neural network also uses multiple memory cells and the output $\hat{Y}$ is calculated using a fully connected layer. Previous studies suggest that GRU performance is similar to LSTM and which network is superior depends on the specific dataset used [9].

Multivariate Linear Regression. Linear regression (LR) is a predictive modelling method which assumes a linear relationship between input and output. Linear regression models are often used as performance baselines for many prediction tasks due to their relatively low complexity and reasonable accuracy in some cases [16, 19]. In this paper, multivariate linear regression models are be used, since traditional LR can only use a single input variable for prediction. Because LR can only predict a single value at a time, a separate regressor is trained for each individual forecasted value $T_{ind}(t+i), i \in [1, h]$ in the predicted horizon. Output of the model

is calculated by passing the input to each of the trained regressors and then collecting their results into the output vector $\hat{Y}$.

SARIMAX. Autoregressive Integrated Moving Average (ARIMA) is a classical, widely used and explored time series forecasting model first introduced more than 4 decades ago [5, 27]. SARIMAX is an extension of classical ARIMA models, improving on them by adding additional parameters, accounting for repeating trends in the time series. Also, unlike ARIMA, SARIMAX are multivariate models, capable of using more than one input variable to predict the output [10, 32]. SARIMAX models have several parameters which define the relationship between input and output. Non-negative integer parameters $p, d, q \in \mathbb{Z}$ define the relationship between past observations and the output. Non-negative parameters $P, D, Q, m \in \mathbb{Z}$, known as seasonal terms, define the seasonal trends within time series as well as the number of time steps in a single season. Traditionally, optimal parameter values are found by manually analysing the time series with statistical methods [10]. However, modern time series forecasting libraries automate this process by performing an parameter search without requiring human intervention [27].

5.2 Metrics for quantitative evaluation

Several metrics have been selected for quantitative evaluation of indoor temperature predictive models trained during experiments. Widely used Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics have been selected for evaluating average predictive model performance over the test set [7]. Maximum error metric (MAXE) has been selected to showcase the worst-case predictive error of the model.

$$RMSE(Y, \hat{Y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{7}$$

$$MAE(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{8}$$

$$MAXE(Y, \hat{Y}) = \max(|Y - \hat{Y}|) \tag{9}$$

where y and $\hat{y}$ are ground truth and predicted one-dimensional vectors, y_i is the i-th element in the vector (starting from 1), n is the number of elements in y and $\hat{y}$.

6 FLEXIBILITY PREDICTION

Heat pump flexibility will be predicted by using *TEMP_FORECAST* to forecast indoor temperature $\hat{Y}$ and calculating how long can the heat pump be in mode *ForcedOn* or mode *Off* before violating user predefined indoor temperature comfort bounds T_{ind_lower} and T_{ind_upper}.

6.1 Prediction method

In this paper, flexibility will be predicted in two scenarios *Off* and *ForcedOn*, as described in Section 4. To predict flexibility in either scenario, the control signal U has to be appropriately defined before forecasting temperature with *TEMP_FORECAST*. When predicting flexibility for *Off*, U should consist of values P_{hp_min}, where P_{hp_min} is the average power consumption of the heat pump being in idle. In most cases, $P_{hp\_min} > 0$ and can be determined using the specification of the heat pump. In this case, $Model(X, U)$ forecasts indoor temperature $\hat{Y}_{off}$ if the heat pump will remain in mode *Off* for prediction horizon h time steps. When predicting flexibility for *ForcedOn*, U should consist of values P_{hp_max}, where P_{hp_max} is the average power consumption of the heat pump operating at its maximum capacity, usually below maximum rated power. In this case, $Model(X, U)$ forecasts indoor temperature $\hat{Y}_{on}$ if the heat pump will remain in mode *ForcedOn* for prediction horizon h time steps.

Algorithm 1 Flexibility prediction algorithm *FLEX_PREDICT*

INPUT:
$\hat{Y}$ - forecasted indoor temperature
T_{ind_lower} - lower bound of temperature
T_{ind_upper} - upper bound of temperature
ε - absolute temperature forecasting error used for adjustment
OUTPUT:
$\hat{F}$ - predicted flexibility

```
procedure FLEX_PREDICT
    F̂ ← 0
    N ← length(Ŷ)
    for i ← 1 to N do
        if (T_ind_lower + ε) ≤ Ŷ(i) ≤ (T_ind_upper − ε) then
            F̂ ← F̂ + 1          ▷ If temperature is within bounds
        else
            break               ▷ If temperature violates bounds
    return F̂
```

Once indoor temperature $\hat{Y}$ is forecasted, flexibility will be predicted using *FLEX_PREDICT*, as defined in Algorithm 1, using $\hat{Y}$ and predefined user temperature bounds T_{ind_lower} and T_{ind_upper}. Optionally, an absolute forecasting error parameter ε, where $\varepsilon >= 0$ can be used to adjust for average *TEMP_FORECAST* error. As seen in Algorithm 1, ε effectively shrinks the temperature bounds by subtracting ε from the upper bound T_{ind_upper} and adding ε to lower bound T_{ind_lower}. If $\varepsilon > 0$, this results in a more pessimistic prediction of potential heat pump flexibility. If error adjustment is not preferred, $\varepsilon = 0$ is used.

6.2 Metrics for evaluating flexibility

In this paper, we propose several metrics for evaluating flexibility prediction performance. Flexibility prediction performance can be numerically evaluated by comparing predicted flexibility $\hat{F}$ calculated using forecasted indoor temperature $\hat{Y}$ and potential flexibility F calculated using ground-truth indoor temperature Y.

Mean Absolute Flexibility Error (MAFE) is an adaptation of MAE for evaluating average absolute error for predicted flexibility.

$$MAFE(Z, \hat{Z}) = \frac{1}{n}\sum_{i=1}^{n}|Z_i - \hat{Z}_i| \tag{10}$$

where $Z = \langle F_1, F_2, \ldots, F_n \rangle$ and $\hat{Z} = \langle \hat{F}_1, \hat{F}_2, \ldots, \hat{F}_n \rangle$ are vectors of predicted flexibility from predicted and ground-truth temperatures, respectively. n is the number of elements in Z and $\hat{Z}$.

Two types of errors are possible while predicting flexibility: underestimation when $\hat{F} < F$ (predict that the heat pump can operate

in a given mode for less time than it possibly could) and overestimation when $\hat{F} > F$ (predict that the heat pump can operate in a given mode for more time than it possibly could). To capture what type of error is prevalent, we introduce two specialized variants of MAFE. Mean Absolute Overestimated Flexibility Error (MAOFE) is a variant of MAFE, which only calculates overestimated flexibility error ($\hat{F} > F$).

$$MAOFE(Z, \hat{Z}) = \frac{1}{n} \sum_{i=1}^{n} |min(Z_i - \hat{Z}_i, 0)| \tag{11}$$

Mean Absolute Underestimated Flexibility Error (MAUFE) only calculates underestimated flexibility error ($\hat{F} < F$).

$$MAUFE(Z, \hat{Z}) = \frac{1}{n} \sum_{i=1}^{n} |max(Z_i - \hat{Z}_i, 0)| \tag{12}$$

Notably, overestimation and underestimation are the only two types of possible errors and are mutually exclusive, meaning that:

$$MAUFE + MAOFE = MAFE \tag{13}$$

Extracted Potential Flexibility Ratio is a metric that shows how much flexibility has been predicted in comparison to how much was available. Notably, this metric does not penalize overestimations, only underestimations.

$$EPFR(Z, \hat{Z}) = \frac{\sum_{i=1}^{n} max(Z_i - \hat{Z}_i, 0)}{\sum_{i=1}^{n} Z_i} \tag{14}$$

These 4 proposed metrics enable to quantitatively evaluate overall flexibility prediction performance, identify what type of error is more prevalent, and measure what percentage of total potential flexibility has been predicted.

7 EXPERIMENTS

In this paper, the experiments were performed using data from 3 heat pump deployments from 2 open datasets containing historical indoor, outdoor temperature and power consumption data.

7.1 Datasets

The first dataset (*NIST Net-Zero*) is collected from the Net-Zero Energy Residential Test Facility, a residential building test bed constructed to study and demonstrate various technologies for effectively meeting typical residential demands using renewable energy. The data were collected from an experiment simulating realistic consumption and usage patterns of a family of four. During the demonstration period, heating for the household was performed by an air-source (air-to-air) heat pump. The heat pump was configured to maintain two constant temperature set points of $23.8°C$ during the cooling season and $21.1°C$ during the heating season. The indoor temperature fluctuated very little throughout the year, changing by less than $2°C$ from the set point. The dataset collected from this experiment includes 1 minute granularity indoor time series, outdoor temperature, and heat pump power consumption readings for one year. This dataset is open access and has been widely used in other studies [28, 36].

The second and third datasets are made available by the New York State Energy Research and Development Authority (NYSERDA). They provide data from 50 geothermal heat pumps installed in residential buildings ranging from around 100 to nearly 600 square meters in footage in New York State in the United States. The two datasets were collected from deployments with identifiers S40 and S44. No set point temperature information was provided, however, in the NYSERDA S40 dataset, the heat pump maintained an indoor temperature of around $21.1°C$ and in the NYSERDA S44 dataset – around $20°C$. The indoor temperature in both sites fluctuated significantly, especially during the winter season, where the indoor temperature in both sites would change by more than $4°C$ from the approximated set points. Both datasets include 15 minute granularity time series indoor, outdoor and power consumption readings over approximately 12 months and are available for open access.

7.2 Data preprocessing

All datasets in the experiments (as defined in Section 7.1) were preprocessed in the following steps:

Data collection. Data for experiments were collected by selecting indoor temperature T_{ind}, outdoor temperature T_{out} and power P_{hp} readings from the datasets. T_{ind} values for NIST Net-Zero dataset were calculated by calculating the mean of all provided indoor temperature readings in the original dataset. Data from two sites were included in the experiments from the NYSERDA dataset: S40 and S44. Furthermore, temperature readings in NYSERDA S40 and NYSERDA S44 were converted from Fahrenheit to Celsius.

Poor quality data removal. Outlier values were removed by selecting data points where temperatures $T_{ind} \in [10; 30]$, $T_{out} \in [-50; 50]$. Furthermore, records with changes of more than $15°C$ within a 15 minute period were also removed. Data marked as incorrect or inconsistent in the provided data specifications were also removed.

Data resampling. In the experiments, 15 minute data granularity was selected to be used across all datasets, as it is usually used in electricity and flexibility markets [15]. For NIST Net-Zero dataset, data were downsampled by averaging power P_{hp} readings to 15 minute intervals, and taking the last outdoor temperature T_{out} and indoor temperature T_{ind} readings. No resampling was required for NYSERDA S40 or NYSERDA S44 datasets.

Date and time feature extraction. As used predictive models require inputs and outputs to have a numeric type, the given date and time timestamp $DT(t) \in X(t)$ was converted into respective year, month, day, hour, minute, day of the week integer values. This was done for each vector $X(t)$ in given historical data X.

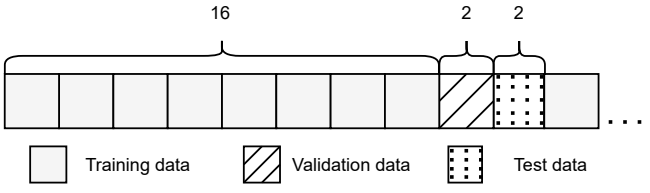

Figure 3: Visual representation of the sequential dataset split strategy applied in intervals of 20 days

Data splitting for training, validation and testing. Each dataset was split into training, validation, and test subsets (sets). Training and

Table 2: Snapshot of the overall best performing indoor temperature forecasting models

	Dataset								
	NIST			NYSERDA S40			NYSERDA S44		
Model	**RMSE**	**MAE**	**MAXE**	**RMSE**	**MAE**	**MAXE**	**RMSE**	**MAE**	**MAXE**
LSTM (c=16, m=16)	0.3240	0.3209	0.3680	**0.2352**	**0.2125**	**0.3266**	**0.3084**	**0.2727**	**0.4448**
LSTM (c=32, m=96)	**0.0833**	**0.0752**	**0.1202**	0.2408	0.2169	0.3357	0.3106	0.2749	0.4475
LSTM average	0.3967	0.3924	0.4397	0.2580	0.2340	0.3570	0.3281	0.2938	0.4653
GRU (c=16, m=16)	0.3200	0.3157	0.3726	0.2513	0.2302	0.3378	0.3171	0.2850	0.4446
GRU (c=16, m=96)	0.1262	0.1209	0.1607	0.2458	0.2230	0.3404	0.3283	0.2936	0.4687
GRU average	0.2835	0.2749	0.3449	0.2780	0.2520	0.3860	0.3544	0.3198	0.4970
LR (m=32)	0.2620	0.2519	0.3328	0.2852	0.2549	0.4053	0.4066	0.3577	0.5884
LR (m=16)	0.3274	0.3240	0.3773	0.2604	0.2281	0.3797	0.4119	0.3622	0.5940
LR average	0.2712	0.2591	0.3431	0.3110	0.2740	0.4560	0.4850	0.4167	0.7365
SARIMAX	0.1424	0.1280	0.2063	0.9086	0.8439	1.1724	2.2577	2.0835	2.8021

validation sets were used during predictive model training process, to learn model weights. The test set was only used to assess model performance on previously unseen data and was not used during model training. All datasets are time series and possess sequential dependencies, trends and seasonality. As such, traditionally used splitting techniques such as k-fold splitting are not applicable.

To account for this, a sequential data split strategy was utilized. The original dataset was divided into parts of one day each and then sequentially divided into three bins for each of the training, validation and test sets. Following best machine learning practices, every 20 days in the dataset, a sequence of 16 days was added to the training set, following a sequence of 2 days which was added to the validation set, after which a sequence of 2 days was added to the test set (as seen in Figure 3). This was repeated until the entire dataset was split. This splitting technique ensures that each set is representative of intra-day temperature change and power consumption patterns and how they change between different months and seasons.

7.3 Implementation details

Grid search was used to find appropriate machine learning model hyperparameter configurations. For LSTM and GRU models, different cell counts $c \in \langle 16, 32, 48, 80 \rangle$ were used. As previous studies have shown that memory parameter m selection can have a significant impact on the performance of the model, several configurations of $m \in \langle 16, 32, 48, 96 \rangle$ were tested with all trained models [37]. During experiments, models were trained with horizon $h = 4$, meaning that the models predicted indoor temperature for the next hour in 4 time steps. Training of all models, apart from SARIMAX, was conducted using the Adam optimizer with learning rate parameter $\alpha = 0.005$, using RMSE as the loss function. Models were trained with a batch size of 64 for a maximum of 125 epochs with early stopping enabled to prevent overfitting. During training and validation losses were logged into persistent storage. Model checkpoints were made every 5 epochs. Checkpoint of the lowest validation loss model was kept in storage. To account for potential outlier results due to the random initial parameter initialization in LSTM, GRU and LR models, each model configuration was trained 3 times and the model with the second-lowest error was used in results analysis and flexibility prediction. For SARIMAX models, optimal parameters were found using the auto-ARIMA parameter search mechanism, finding parameters p, d, q, P, D, Q, m.

Dataset preprocessing, analysis, flexibility prediction, result collection and visualization was realized using the PyData data analysis toolkit in Python. LSTM, GRU, and linear regression indoor temperature predictive models were implemented, trained and tested using the open source machine learning library PyTorch (version 1.6.0) and high level API extension library PyTorch Lightning (version 1.1.1). SARIMAX models were fitted and tested using open source library Pmdarima (version 1.8.0). Model training and evaluation was carried out on several workstations equipped with NVIDIA graphics processing units and using the Google Colab platform.

8 RESULTS

During the experiments, the indoor temperature forecasting models were trained using the training set and validation set. The indoor temperature forecasting and flexibility prediction error was then evaluated using the test set. Each model was trained and evaluated using data from only one dataset.

8.1 Indoor temperature forecasting results

Quantitative model performance. Trained indoor temperature forecasting model performance was quantitatively evaluated using data from the test set by calculating average *RMSE*, *MAE*, and *MAXE* metric values. Results from the experiments indicate that the trained predictive models managed to effectively capture how indoor temperature changes w.r.t. heat pump operation and outdoor temperature. Quantitative model performance evaluation shows that the best performing predictive models were trained using the LSTM and GRU neural network architectures, with LSTM having the overall best performance. As seen in Table 2, best performing LSTM and GRU architecture models outperform multivariate linear regression and SARIMAX models. LSTM models have the lowest error in experiments on all three deployments, outperforming GRU by over 9% lower RMSE, multivariate LR models by over 32% lower RMSE. Best performing LSTM models also significantly outperform

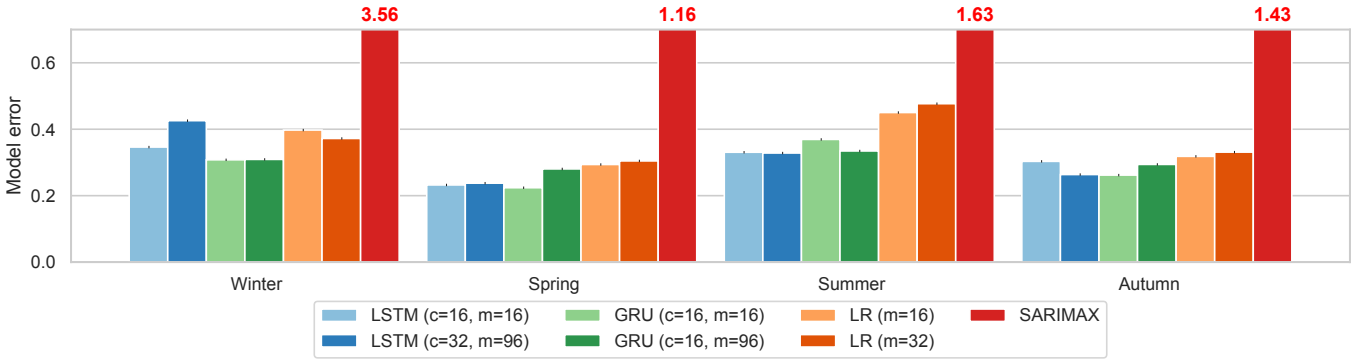

Figure 4: Best performing indoor temperature forecasting model RMSE error by season on NYSERDA S44 dataset

SARIMAX models, especially on NYSERDA S40 and NYSERDA S44 datasets, where LSTM models recorded over 82% lower RMSE error.

Quantitative model results also show that baseline linear regression and SARIMAX models fitted with auto-ARIMA have reasonable performance on the NIST Net-Zero dataset, where the indoor temperature does not change by more than a few degrees over the entire year. As seen in Table 2 SARIMAX outperforms the best performing LR model by over 45% lower RMSE on the NIST Net-Zero dataset. However, these baseline models have significantly higher predictive error on NYSERDA S44 than their LSTM and GRU counterparts. Multivariate LR models have, on average, over 30% higher error and SARIMAX has over 7 times higher error than GRU or LSTM models. These significant performance differences suggest that SARIMAX and multivariate linear regression models, although generally effective, are not capable of effectively adapting to rapid indoor temperature fluctuations and capturing other complex patterns present in the NYSERDA S44 dataset.

Seasonal performance analysis. After quantitative evaluation of trained indoor temperature predictive models, model performance was also analysed across different seasons. This indicates whether trained models are effective at capturing different patterns present throughout the entire calendar year. As all of the heat pump deployments are situated in the northern hemisphere, we use 4 meteorological seasons of winter, spring, summer and autumn, spanning December to February, March to May, June to August, and September to November, respectively.

Seasonal analysis showed that LSTM and GRU predictive models have high accuracy during the entire year, with their predictive error changing by less than 10% between different seasons. However, baseline multivariate linear regression and SARIMAX model performance vary significantly between seasons. This trend is the most pronounced in the experiments conducted on the NYSERDA S44 dataset. As seen in Figure 4, best performing LSTM and GRU models have similar predictive errors across seasons on the NYSERDA S44 dataset. However, multivariate LR and SARIMAX model performance fluctuate, with linear regression having 30% higher error in summer season and around 15% higher error in winter season and SARIMAX having more than 9 times worse error during the winter season than the best performing models. Poor performance during the colder seasons is highly undesirable, as heat pumps operate and use more power when the outside temperatures are lower, increasing the amount of potential flexibility [15]. Intuitively, the difference of performance between seasons can be attributed to the fact that both multivariate linear regression and SARIMAX models have a very limited amount of trainable parameters and thus are unable to capture all of the emerging patterns between seasons.

Model configuration comparison. Although it has been established that LSTM and GRU models possess high performance in terms of numerical error, some model configurations produced comparatively poor results. LSTM and GRU models trained using cell count $c = 80$ had significantly worse results on average than best performing models trained with $c = 16$ or $c = 32$. On the NIST Net-Zero dataset models with cell count configuration $c = 80$ recorded average RMSE +0.5 higher, compared to the best performing model LSTM (c=32, m=96) and, on average, +0.6 higher on the NYSERDA S44 and the NYSERDA S40 datasets. Inspection of validation loss during training indicated that some of these models failed to converge during training, even across multiple runs, despite having potentially higher capacity of capturing complex trends due to the increase in cell count. These results along with quantitative results presented in Table 2 indicate cell count hyperparameter selection is crucial for training accurate LSTM and GRU models.

Model training time. In order to evaluate model training time fairly, all models were trained using the same Google Colab instance with an NVIDIA Tesla P100 GPU. This includes dataset loading, model checkpointing, logging, generating, and saving predictions on the test set. All indoor temperature predictive models were trained in less than 10 minutes on the given Colab instance. Prediction generation using an already trained model takes $\leq 10ms$ using the Google Colab instance and Apple MacBook Pro with an Intel i5 2.0 GHz CPU.

8.2 Flexibility prediction results

Heat pump flexibility prediction was performed using trained indoor temperature predictive models (as shown in Figure 1). Heat pump flexibility scenarios *Off* and *ForcedOn* were examined, coinciding with SG-Ready heat pump operation modes *Off* and *ForcedOn* (as defined in Table 1).

For scenario *Off*, data for flexibility prediction performance evaluation were collected from the test set by taking records where heat pump power was below a predefined power threshold for at least 45 minutes from start. The threshold of $100W$ for mode *Off*

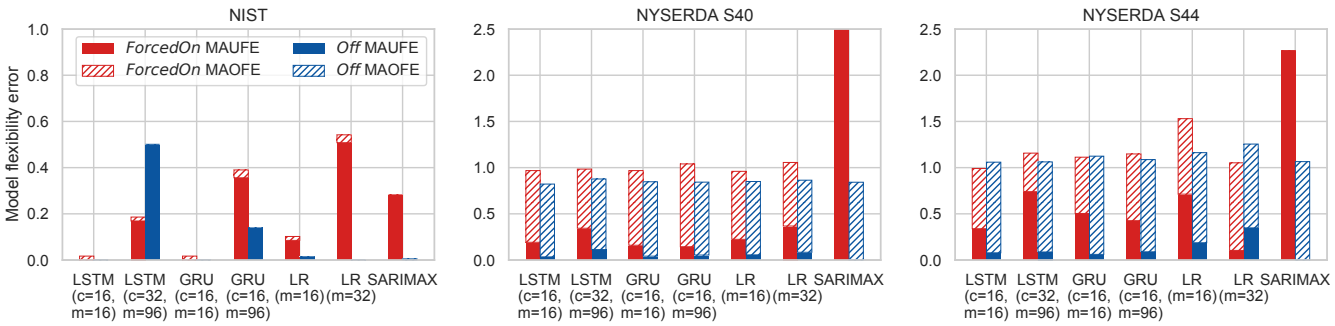

Figure 5: Flexibility prediction results using best performing models from Table 2 with user temperature threshold of $0.5°C$

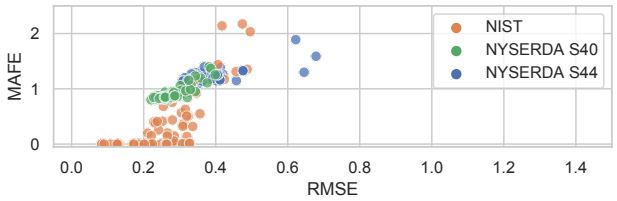

Figure 6: Diagram visualizing MAFE and RMSE correlation of all the trained models during experiments

was used for NIST Net-Zero, NYSERDA S40, and NYSERDA S44 datasets.

For scenario *ForcedOn*, data for flexibility prediction performance evaluation were also collected from the test set. Records were selected, where heat pump power was above a predefined power threshold for at least 45 minutes from start. The threshold for mode *ForcedOn* was $1.5kW$ for NIST Net-Zero dataset, $2.14kW$ for NYSERDA S44 dataset, and $0.92kW$ for the NYSERDA S40 dataset.

Flexibility was predicted using *FLEX_PREDICT* defined in Algorithm 1. Lower and upper bounds $T_{ind\_lower}, T_{ind\_upper}$ were calculated by adding and subtracting an indoor temperature threshold value from the last known indoor temperature to get upper and lower bounds, respectively. $0.5°C$ was selected as the indoor temperature threshold value. No adjustment for predictive model error was used in initial experiments ($\varepsilon = 0$).

Quantitative flexibility prediction performance. Flexibility prediction using trained predictive models experiments show that HeatFlex is capable of predicting flexibility with a high degree of accuracy. As seen in Figure 5, best performing trained temperature predictive models perform well across all three datasets.

For the NIST Net-Zero dataset, the average MAFE error for scenario *Off* was 0.1, with underestimated flexibility being the only type of error. For scenario *ForcedOn*, MAFE was 0.28, with underestimated error MAUFE constituting for over 95% of total error. The best performing model for flexibility prediction was the GRU (c=16, m=16), followed closely by LSTM (c=16, m=16). Notably, both GRU (c=16, m=16) and LSTM (c=16, m=16) had more than 3 times higher RMSE indoor temperature predictive error than the best performing indoor temperature predictive model LSTM (c=32, m=96), which had the worst total MAFE for both modes across all 7 models. This indicates that temperature predictive model RMSE does not always translate into better flexibility prediction MAFE error.

For NYSERDA S40 dataset, the average MAFE error for scenario *Off* was 0.8 with overestimated flexibility constituting over 95% of total error. All of the models performed very similarly, predicting flexibility within 5% of the average 0.8. The average MAFE for *ForcedOn*, excluding the SARIMAX model results, was 0.98. SARIMAX underperformed significantly with MAFE of 2.5. The best overall performing model for flexibility prediction was the GRU (c=16, m=16).

For NYSERDA S44 dataset, the average MAFE error for scenario *Off* was 1.1. Similarly to the results on the NYSERDA S40 dataset overestimated flexibility made up over 90% of the total error. For predicting flexibility for *ForcedOn*, the average error was 1.35, with overestimated flexibility error MAOFE constituting for around 60% of the total error. The best performing model for flexibility prediction was the LSTM (c=16, m=16), which was also the best performing temperature predictive model (as seen in Table 2).

Correlation analysis of predictive model error and predicted flexibility. Trained model flexibility prediction results were also analysed to identify if a numeric relationship exists between indoor temperature predictive model error and predicted flexibility error. Figure 6 visualizes the bivariate correlation between trained model RMSE and the mean MAFE for scenarios *Off* and *ForcedOn*. The figure shows that in models of all three datasets there is a positive correlation between RMSE and MAFE. However, the correlation is non-linear and has a lot of outliers, meaning that indoor temperature RMSE is a reasonable metric for estimating flexibility error MAFE, but higher indoor temperature model accuracy does not always translate into more accurate flexibility prediction.

Adjustment for predictive model error results. In HeatFlex, indoor temperature prediction and flexibility prediction is separated, allowing adjusting for trained indoor temperature predictive model error during flexibility prediction. One adjustment strategy is to use the ε adjustment variable in the flexibility prediction algorithm (as defined in Algorithm 1). If $\varepsilon > 0$, it reduces the range between user defined comfort bounds. As seen in Figure 7 the user comfort bound reduction forces the predictive models to underestimate flexibility, reducing MAOFE error at the cost of increasing MAUFE, which might be a preferable trade-off, depending on the use-case. An alternative strategy for adjusting for predictive model error is to widen the user comfort bounds by negotiating with the end-user. It could

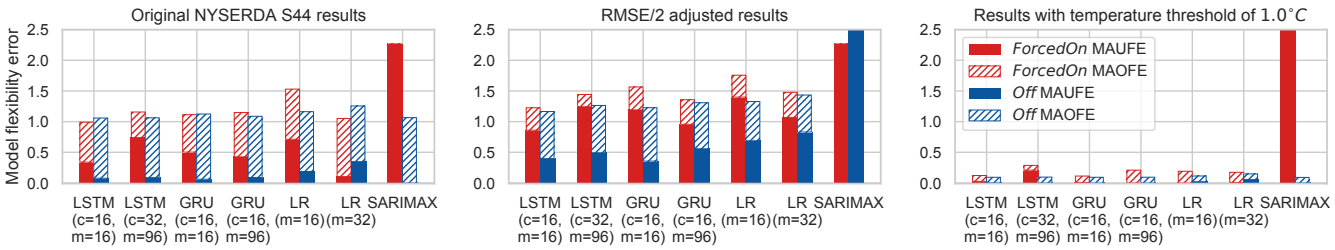

Figure 7: Flexibility prediction results for NYSERDA S44 dataset using indoor temperature threshold of $0.5^{\circ}C$ *(left)* compared with adjustment strategy RMSE/2 *(center)* and increased temperature threshold of $1.0^{\circ}C$ *(right)*

Table 3: Total flexible energy (kWh) and EPFR percentage of best performing models

	NIST		NYSERDA S40		NYSERDA S44	
	mode = Off	*mode = ForcedOn*	*mode = Off*	*mode = ForcedOn*	*mode = Off*	*mode = ForcedOn*
Flexible Energy (E_{mode})	1166 kWh	430 kWh	1087 kWh	714 kWh	1855 kWh	785 kWh
EPFR	98.70%	87.20%	97.30%	86.20%	87.40%	95.40%

be used in cases where previous adjustment for error strategies do not yield satisfactory results. Widening the comfort bounds has a two-fold effect: it potentially reduces the flexibility prediction error and increases the total amount of potential heat pump flexibility. As seen in Figure 7, the widening of the user comfort bounds by $0.5^{\circ}C$ decreased MAFE by 85% on average. The widening of bounds also yielded an over 41% increase in potential flexibility for modes *Off* and *ForcedOn*.

Evaluation of the amounts of predicted flexible energy. After all experiments were completed, flexibility prediction results were aggregated and the total amount of flexible energy was calculated. To calculate flexible energy for $mode \in \{ForcedOn, Off\}$, the average power consumptions of the heat pump were calculated in both normal operation $\overline{P_{normal}}$ and in the given mode $\overline{P_{mode}}$ and then the absolute difference was multiplied the predicted flexibility amount $\hat{F}_{mode}$ divided by 4 to normalize to hours (one time step every 15 minutes).

$$E_{mode} = (\overline{P_{mode}} - \overline{P_{normal}}) * \frac{\hat{F}_{mode}}{4} \tag{15}$$

The comfort bound of $0.5^{\circ}C$ was used across all datasets for flexibility prediction. As seen in Table 3 all of the heat pump deployments provide over $1.5MWh$ of combined flexible energy, with the NYSERDA S44 providing $2.64MWh$. The results indicate that the majority (over 68%) of total potential flexible energy can be utilized with scenario *Off*. On average, the flexibility prediction models managed to predict over 90% of available flexibility (EPFR) from all three datasets between both modes. Notably, all of the monitored heat pumps used in this study were not configured to maximize flexible energy and were operating to maintain a set temperature.

9 CONCLUSIONS AND FUTURE WORK

This paper proposes HeatFlex, a novel data-driven method for predicting flexibility from smart heat pumps. HeatFlex, utilizing modern machine learning models, LSTM and GRU recurrent neural networks, can accurately predict flexibility from heat pump devices without requiring knowledge about deployed heat pump parameters, building layout or environment details or manual development of physical models. Furthermore, HeatFlex only uses three, widely available monitored variables, making it applicable in a majority of use cases. The paper also proposes new metrics for heat pump flexibility, enabling quantitative evaluation of flexibility prediction performance. Conducted experiments on 3 open access datasets show that machine learning based indoor temperature forecasting models can be trained in a scalable, data-driven way and be effective at predicting energy flexibility from individual smart heat pump devices. Quantitative and qualitative performance evaluation shows that HeatFlex, utilizing recurrent neural networks, has over 32% lower indoor temperature forecasting error than the model baselines linear regression and SARIMAX, while also having more consistent performance throughout different seasons. Experimental results show that HeatFlex can be effective in different deployments, with different building sizes and heat pump types, predicting over 90% of potential heat pump flexibility.

In future work, we will develop data-driven predictive models that could predict flexibility from other types of devices (e.g. air conditioners, electric vehicle chargers). We will perform experiments using additional data from real smart heat pump deployments as well as simulations to verify HeatFlex robustness in various flexible operation conditions. Additionally, we will explore developing predictive models that optimize directly on flexibility prediction error, thus maximizing utility. Finally, we will develop novel probabilistic models which will enable evaluating potential prediction accuracy during inference.

ACKNOWLEDGMENTS

This work was supported by the FEVER (Flexible Energy Production, Demand and Storage-based Virtual Power Plants for Electricity Markets and Resilient DSO Operation) project under the Horizon 2020 programme.

REFERENCES

[1] Alessandro Aliberti, Francesca Maria Ugliotti, Lorenzo Bottaccioli, Giansalvo Cirrincione, Anna Osello, Enrico MacIi, Edoardo Patti, and Andrea Acquaviva. 2018. Indoor Air-Temperature Forecast for Energy-Efficient Management in Smart Buildings. *Proceedings - 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe, EEEIC/I and CPS Europe 2018* (2018). https://doi.org/10.1109/EEEIC.2018.8494382

[2] Abdulaziz Almalaq and George Edwards. 2017. A review of deep learning methods applied on load forecasting. *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017* 2017-December (2017), 511–516. https://doi.org/10.1109/ICMLA.2017.0-110

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 5, 2 (1994), 157–166. https://doi.org/10.1109/72.279181

[4] Matthias Boehm, Lars Dannecker, Andreas Doms, Erik Dovgan, Bogdan Filipič, Ulrike Fischer, Wolfgang Lehner, Torben Bach Pedersen, Yoann Pitarch, Laurynas Šikšnys, and Tea Tušar. 2012. Data management in the MIRABEL smart grid system. *ACM International Conference Proceeding Series* (2012), 95–102. https://doi.org/10.1145/2320765.2320797

[5] George Edward Pelham Box and Gwilym Jenkins. 1990. *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc., USA.

[6] P. Carroll, M. Chesser, and P. Lyons. 2020. Air Source Heat Pumps field studies: A systematic literature review. *Renewable and Sustainable Energy Reviews* 134, August (2020). https://doi.org/10.1016/j.rser.2020.110275

[7] T. Chai and R. R. Draxler. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development* 7, 3 (2014), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

[8] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2015. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. (2015), 103–111. https://doi.org/10.3115/v1/w14-4012 arXiv:1409.1259

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. (dec 2014). arXiv:1412.3555 http://arxiv.org/abs/1412.3555

[10] Tingting Fang and Risto Lahdelma. 2016. Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system. *Applied Energy* 179 (2016), 544–552. https://doi.org/10.1016/j.apenergy.2016.06.133

[11] David Fischer, Marc Andre Triebel, and Oliver Selinger-Lutz. 2018. A Concept for Controlling Heat Pump Pools Using the Smart Grid Ready Interface. *Proceedings - 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2018* (2018). https://doi.org/10.1109/ISGTEurope.2018.8571870

[12] Benedikt Heidrich, Marian Turowski, Nicole Ludwig, Ralf Mikut, and Veit Hagenmeyer. 2020. Forecasting energy time series with profile neural networks. In *e-Energy 2020 - Proceedings of the 11th ACM International Conference on Future Energy Systems*. Association for Computing Machinery, Inc, New York, NY, USA, 220–230. https://doi.org/10.1145/3396851.3397683

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[14] Chiou Jye Huang and Ping Huan Kuo. 2019. Multiple-Input Deep Convolutional Neural Network Model for Short-Term Photovoltaic Power Forecasting. *IEEE Access* 7 (2019), 74822–74834. https://doi.org/10.1109/ACCESS.2019.2921238

[15] Konstantinos Kouzelis, Zheng H. Tan, Birgitte Bak-Jensen, Jayakrishnan Radhakrishna Pillai, and Ewen Ritchie. 2015. Estimation of Residential Heat Pump Consumption for Flexibility Market Applications. *IEEE Transactions on Smart Grid* 6, 4 (2015), 1852–1864. https://doi.org/10.1109/TSG.2015.2414490

[16] Guokun Lai, Wei Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long- and short-term temporal patterns with deep neural networks. *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018* July (2018), 95–104. https://doi.org/10.1145/3209978.3210006 arXiv:arXiv:1703.07015v3

[17] Ming Li, Yijun Li, and Xinli Min. 2020. Practice and Application of LSTM in Temperature Prediction of HVAC System. *Proceedings of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020* Itoec (2020), 1000–1004. https://doi.org/10.1109/ITOEC49072.2020.9141910

[18] Junyu Liu, Xiao Wang, Yan Zhao, Bin Dong, Kuan Lu, and Ranran Wang. 2020. Heating Load Forecasting for Combined Heat and Power Plants Via Strand-Based LSTM. *IEEE Access* 8 (2020), 33360–33369. https://doi.org/10.1109/ACCESS.2020.2972303

[19] Fernando Mateo, Juan José Carrasco, Abderrahim Sellami, Mónica Millán-Giraldo, Manuel Domínguez, and Emilio Soria-Olivas. 2013. Machine learning methods to forecast temperature in buildings. *Expert Systems with Applications* 40, 4 (2013), 1061–1068. https://doi.org/10.1016/j.eswa.2012.08.030

[20] Bijay Neupane, Torben Bach Pedersen, and Bo Thiesson. 2018. Utilizing device-level demand forecasting for flexibility markets. *e-Energy 2018 - Proceedings of the 9th ACM International Conference on Future Energy Systems* (2018), 108–118. https://doi.org/10.1145/3208903.3208922

[21] Bijay Neupane, Laurynas Siksnys, and Torben Bach Pedersen. 2017. Generation and evaluation of flex-offers from flexible electrical devices. In *e-Energy 2017 - Proceedings of the 8th International Conference on Future Energy Systems*. Association for Computing Machinery, Inc, 143–156. https://doi.org/10.1145/3077839.3077850

[22] Thomas Nowak and Pascal Westring. 2017. Growing for good ? The European Heat Pump Market - Status and outlook. *12th IEA Heat Pump Conference 2017* (2017), 1–10.

[23] Dimitrios Papadaskalopoulos, Goran Strbac, Pierluigi Mancarella, Marko Aunedi, and Vladimir Stanojevic. 2013. Decentralized participation of flexible demand in electricity markets - Part II: Application with electric vehicles and heat pump systems. *IEEE Transactions on Power Systems* 28, 4 (2013), 3667–3674. https://doi.org/10.1109/TPWRS.2013.2245687

[24] Debayan Paul, Tanmay Chakraborty, Soumya Kanti Datta, and Debolina Paul. 2018. IoT and Machine Learning Based Prediction of Smart Building Indoor Temperature. *2018 4th International Conference on Computer and Information Sciences: Revolutionising Digital Landscape for Sustainable Smart Society, ICCOINS 2018 - Proceedings* (2018), 1–6. https://doi.org/10.1109/ICCOINS.2018.8510597

[25] Torben Bach Pedersen, Thibaut Le Gully, Per D. Pedersen, Luis L. Ferreira, Laurynas Šikšnys, Petr Stluka, Michele Albano, Arne Skou, and Petur Olsen. 2016. An Energy Flexibility Framework on the Internet of Things. 2 (2016), 17–37. https://doi.org/10.5220/0006163400170037

[26] Torben Bach Pedersen, Laurynas Siksnys, and Bijay Neupane. 2018. Modeling and Managing Energy Flexibility Using FlexOffers. *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2018* (2018). https://doi.org/10.1109/SmartGridComm.2018.8587605

[27] Rob J Hyndman and Yeasmin Khandakar. 2008. Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software* 27, 3 (2008), 22. http://www.jstatsoft.org/v27/i03/paper

[28] Olga Rybnytska, Laurynas Šikšnys, Torben Bach Pedersen, and Bijay Neupane. 2020. PGFMU: Integrating data management with physical system modelling. *Advances in Database Technology - EDBT* 2020-March (2020), 109–120.

[29] Laurynas Šikšnys, Emmanouil Valsomatzis, Katja Hose, and Torben Bach Pedersen. 2015. Aggregating and Disaggregating Flexibility Objects. *IEEE Transactions on Knowledge and Data Engineering* 27, 11 (2015), 2893–2906. https://doi.org/10.1109/TKDE.2015.2445755

[30] Jiancai Song, Guixiang Xue, Xuhua Pan, Yunpeng Ma, and Han Li. 2020. Hourly heat load prediction model based on temporal convolutional neural network. *IEEE Access* 8 (2020), 16726–16741. https://doi.org/10.1109/ACCESS.2020.2968536

[31] Hai Tao, Li Junjie, Shi Yu, Chen Yongjian, and Liu Zhenyu. 2020. Predictive analysis of indoor temperature and humidity based on BP neural network single-step prediction method. *Proceedings of 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education, ICISCAE 2020* (2020), 402–407. https://doi.org/10.1109/ICISCAE51034.2020.9236853

[32] Stylianos I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis. 2016. Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting. *2016 IEEE International Energy Conference, ENERGYCON 2016* (2016), 0–5. https://doi.org/10.1109/ENERGYCON.2016.7514029

[33] Emmanouil Valsomatzis, Katja Hose, Torben Bach Pedersen, and Laurynas Šikšnys. 2015. Measuring and comparing energy flexibilities. *CEUR Workshop Proceedings* 1330, c (2015), 78–85.

[34] Emmanouil Valsomatzis, Torben Bach Pedersen, and Alberto Abelló. 2018. Day-ahead trading of aggregated energy flexibility. *e-Energy 2018 - Proceedings of the 9th ACM International Conference on Future Energy Systems* (2018), 134–138. https://doi.org/10.1145/3208903.3208936

[35] Inna Vorushylo, Patrick Keatley, Nikhilkumar Shah, Richard Green, and Neil Hewitt. 2018. How heat pumps and thermal energy storage can be used to manage wind power: A study of Ireland. *Energy* 157 (2018), 539–549. https://doi.org/10.1016/j.energy.2018.03.001

[36] Wei Wu, Harrison M. Skye, and Piotr A. Domanski. 2018. Selecting HVAC systems to achieve comfortable and cost-effective residential net-zero energy buildings. *Applied Energy* 212, October 2017 (2018), 577–591. https://doi.org/10.1016/j.apenergy.2017.12.046

[37] Keming Yan, Chris Diduch, and Mary E. Kaye. 2019. An improved temperature prediction technique for HVAC units using intelligent algorithms. *2019 IEEE Energy Conversion Congress and Exposition, ECCE 2019* (2019), 490–494. https://doi.org/10.1109/ECCE.2019.8912944